123

AF

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: Hon Wah Chin	Attorney Docket No.: CISCP054
Application No.: 09/164,388	Examiner: Prieto, B.
Filed: September 30, 1998	Group: 2152

Title: REDUCING CPU OVERHEAD IN THE FORWARDING PROCESS

CERTIFICATE OF MAILING
I hereby certify that this correspondence is being deposited with the U.S. Postal Service with sufficient postage as first-class mail on September 14, 2004 in an envelope addressed to the Commissioner for Patents, Mail Stop Appeal Brief-Patent, P.O. Box 1450 Alexandria, VA 22313-1450.

Signed: ______________________
Deborah Neill

TRANSMITTAL OF REPLY BRIEF IN RESPONSE TO EXAMINER'S ANSWER

Mail Stop Appeal Brief-Patents
Commissioner of Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

Transmitted herewith in triplicate is the Reply Brief In Response To Examiner's Answer mailed July 14, 2004.

This reply brief is being filed within two (2) months of the mailing date of the Examiner's Answer.

Applicant believes that no extension of term is required. If an additional extension of time is required, however, please consider this a petition therefor.

☒ Charge any additional fees or credit any overpayment to Deposit Account No. 500388, (Order No. CISCP054).

Respectfully submitted,
BEYER WEAVER & THOMAS, LLP

Elise R. Heilbrunn
Registration No. 42,649

P.O. Box 778
Berkeley, CA 94704-0778
(510) 843-6200

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF APPEALS

EX PARTE HON WAH CHIN

Application for Patent

Filed September 30, 1998

Serial No. 09/164,388

FOR:

REDUCING CPU OVERHEAD IN THE FORWARDING PROCESS

REPLY BRIEF TO EXAMINER'S ANSWER

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as First Class Mail in an envelope addressed to: Commissioner of Patents and Trademarks, P.O. Box 1450, Alexandria, VA 22313-1450 on September 14, 2004.

Signed: ______________________
Deborah Neill

BEYER WEAVER & THOMAS, LLP
Attorneys for Applicant

TABLE OF CONTENTS

I. REAL PARTY IN INTEREST

The real party in interest is Cisco Technology, Inc., the assignee of the present application.

II. RELATED APPEALS AND INTERFERENCES

The undersigned is not aware of any related appeals and/or interferences.

III. STATUS OF THE CLAIMS

There have been a total of 54 claims presented throughout the prosecution of this application. Claims 1-43 and 45-54 are pending and stand rejected in this application and are subject to this appeal. Claim 44 has been cancelled.

Claims 1-43 and 45-54 stand rejected under 35 USC §103(a).

IV. STATUS OF AMENDMENTS

On October 30, 2003, the Examiner entered a final rejection. The applicants previously filed Amendment F on July 14, 2003, which was **entered**.

V. SUMMARY OF INVENTIONS

The present invention relates generally to methods and apparatus for reducing CPU overhead in the forwarding process. More particularly, the present invention relates to forwarding packets in a network device (e.g., router) such that per packet CPU involvement typically required when moving a packet from an inbound interface to an outbound interface is reduced or eliminated. This is accomplished, in part, by supporting the simultaneous transfer of multiple packets (e.g., queue of packets).

As described in Applicant's specification, the traditional router has several potentially undesirable characteristics. First, an inbound port of a router typically has a single inbound queue associated therewith. Second, when a packet is forwarded, a single packet or entry in the inbound queue is transferred by an inbound controller to an outbound controller. Third, when the packet is received by the outbound controller, information associated with a single packet is stored in an entry in an outbound queue. Thus, a substantial amount of CPU overhead is consumed during the forwarding process.

Each of the independent claims provides at least one of the following advantages or limitations over the prior art. First, a plurality of inbound queues is provided for a single inbound port. An inbound packet is therefore classified in one of the plurality of inbound queues to enable the inbound packet to be stored in the appropriate queue. Second, one of the plurality of inbound queues is transferred to an outbound controller and/or outbound queue capable of storing (or identifying) a multiplicity of inbound queues. In other words, a queue of packets rather than a single packet is transferred to the outbound controller and/or an associated outbound queue such that a reference (e.g., pointer) to the queue of packets is stored in a single entry in the outbound queue. In other words, a reference to each of the multiplicity of inbound queues is stored in a different entry in the outbound queue. Third, in some embodiments of the invention, it is possible to encrypt an inbound queue prior to transmission by an outbound controller. One or more of the above-described limitations are present in each of the claims. In this manner, CPU overhead in the forwarding process is reduced.

VI. ISSUES

The issues which applicant believes to be most pertinent to the present appeal include:

(a) Whether the combination of U.S. Patent No. 6,487,212 issued to Erimli (Erimli) and U.S. Patent No. 5,392,401 issued to Barucchi (Barucchi) reasonably suggests an **inbound controller** or inbound controller method wherein the inbound controller determines when one of a plurality of inbound queues is ready to be moved to an entry in an outbound queue. (Claims 1-9, 20-27).

(b) Whether any combination of Erimli, Barucchi, and U.S. Patent No. 5,177,480 issued to Clark (Clark) reasonably suggests an **inbound controller** or inbound controller method wherein the inbound controller determines when one of a plurality of inbound queues is ready to be moved to an entry in an outbound queue. (Claims 45-48, 52)

(c) Whether the combination of Erimli and Barucchi reasonably suggests an **outbound controller** or outbound controller method wherein the outbound controller transfers an inbound queue storing a plurality of packets to one of a plurality of entries in an outbound queue. (Claims 10-18, 28-36).

(d) Whether any combination of Erimli, Barucchi and Clark reasonably suggests an **outbound controller** or outbound controller method wherein the outbound controller transfers an inbound queue storing a plurality of packets to one of a plurality of entries in an outbound queue. (Claim 51).

(e) Whether the combination of Erimli and Barucchi reasonably suggests a **router** or router method in which a packet that is received is classified in one of a plurality of inbound queues associated with an inbound port and wherein one of the plurality of inbound queues storing a plurality of packets is transferred to one of a plurality of outbound queues. (Claim 19)

(f) Whether any combination of Erimli, Barucchi and Clark reasonably suggests a **router** or router method in which a packet that is received is classified in one of a plurality of inbound queues associated with an inbound port and wherein one of the plurality of inbound queues storing a plurality of packets is transferred to one of a plurality of outbound queues. (Claims 53, 54)

(g) Whether the combination of Erimli and Barucchi reasonably suggests a **router** in which a plurality of inbound queues are associated with one of a plurality of inbound ports, a plurality of outbound queues are associated with a plurality of outbound ports, and wherein each of the outbound queues is capable of storing a plurality of inbound queues such that a reference to each of the plurality of inbound queues is stored in a different one of a plurality of entries in the outbound queue. (Claim 37-42)

(h) Whether any combination of Erimli, Barucchi, and U.S. Patent No. 5,177,480 issued to Clark (Clark) reasonably suggests an **encryption system** in which an encryption box is adapted for encrypting one of a plurality of inbound queues received by an outbound controller and wherein the outbound controller includes an outbound classifier adapted for classifying the encrypted inbound queue in one of a plurality of outbound queues and storing a reference to the encrypted inbound queue in a single entry in the one of the plurality of outbound queues. (Claim 43, 49-50)

VII. GROUPING OF CLAIMS

With regard to issues (d) and (e), the Examiner noted that the claims were not grouped in the "Grouping of Claims" section. Since there was only one claim mentioned with respect to issues (d) and (e), respectively, Applicant assumed that it didn't need to be explicitly state that the claim in group (d) would be argued independently, and that the claim in group (e) would be argued independently. In other words, since there was only one claim in each of groups (d) and (e), it seemed unnecessary to state that claim 19 does not stand or fall together (or claim 51 does not stand or fall together). These claims are argued independently, and therefore should be considered independently.

With regard to issue (d), the rejected claim does not stand or fall together, and claim 51 will be argued as a group.

With regard to issue (e), the rejected claim does not stand or fall together, and claim 19 will be argued as a group.

VIII. ARGUMENT

(a) The combination of Erimli and Barucchi neither discloses nor suggests the invention of claims 1-9, 20-27

In response to the Examiner's assertion that "what is stored are packets not queues according to the claim," Applicant respectfully asserts that it is irrelevant whether packets are stored. Moreover, the fact that packets are stored in the inbound (and outbound) queue is not disputed. What is relevant is the manner in which packets are transferred and stored in the outbound queue. Specifically, an entire inbound queue of packets is transferred to an entry in an outbound queue. As a result, a plurality of packets are simultaneously transferred. In contrast, the combination of the cited art discloses transferring a single packet or frame at a time. In the pending claims, it is determined "when one inbound queue storing a plurality of packets is ready to be moved to an entry in an outbound queue." In other words, a plurality of packets are transferred simultaneously (when one of the plurality of inbound queues storing a plurality of packets is ready to be moved to an entry in an outbound queue), since an entire inbound queue storing a plurality of packets is moved to an entry in an outbound queue (when an inbound queue is ready to be moved to an entry in an outbound queue).

In contrast, the Examiner admits in the Examiner's reply brief on page 11 that "Erimli teaches storing an inbound frame packet received at an inbound port in an inbound queue....a pointer referencing the location at which the frame is stored in the inbound queue, the pointer is stored into a particular output queue (or queued) including placing the pointer on the top of the appropriate output queue..." These statements seem to contradict the Examiner's assertion that "[a]rguments that the reference teaches storing pointers in an outbound queue but does not teach queueing queues are not persuasive." Erimli clearly teaches transferring a pointer to a single frame (and therefore a single entry in the inbound queue to an entry in the outbound queue). Accordingly, Erimli teaches away from simultaneously transferring an entire inbound queue to an entry in an inbound queue.

Also worth mentioning is that the Examiner fails to acknowledge the limitation reciting "each of the multiplicity of inbound queues storing a plurality of packets to be separately transmitted." On page 11 of the Examiner's reply brief, the Examiner refers to data that "is accumulated in the inbound queue to then burst, a number of bytes of information at a time," citing col. 10, lines 1-11. However, the Examiner appears to be taking this out of context.

The Examiner asserts that Erimli discloses the bursting of information accumulated in the inbound queue. Col. 9, lines 29-47, state that "the output queue write side 76 receives an entry. In the exemplary embodiment of a multipart switch 12 according to the present invention, the entry is a frame pointer that points to the first buffer in external memory in which the first 240 bytes of a frame are stored...After the entry flows through and reaches the bottom of the output queue write side 76, control logic associated with the output queue 74 makes a decision as to what to do with the entry. If there is space in the output queue read side 78, and the overflow area 110 for that output queue 74 is empty, then one or more entries are passed directly from the output queue write side 76 to the output queue read side." Moreover, col. 9, lines 61-67, discloses that "In the output queue structure, the read side 78 is acting most like a traditional queue, because it is from this portion that entries are taken, one by one. The output queue write side 76 mostly serves a collection function to assemble the data into bursts for writing to the external memory 36." Thus, Erimli discloses transferring separate entries from the output queue write side to the output queue read side. The bursting of data is only performed when data is transferred from the write side 76 to the overflow area 110 in external memory. See col. 10, lines

1-10. Even if Erimli did disclose bursting to transfer data from the output queue write side to the output queue read side, each entry of the outbound queue would still include only a single entry from the inbound queue. Accordingly, Erimli fails to disclose the transferring of an inbound queue to an entry in an outbound queue, or determining when an inbound queue is ready to be moved to an entry in an outbound queue.

It is also important to note that on page 12 of the Examiner's answer, the Examiner states that "In response to applicant's argument that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (ie..."enqueueing an entire queue of packets") are not recited in the rejected claim(s). Applicant respectfully disagrees. In fact, claim 4, for example, specifically recites:

> "The method as recited in claim 1, the selected one of the plurality of inbound queues corresponding to one of a plurality of outbound queues, the method further comprising:
>
> transferring the selected one of the plurality of inbound queues storing a plurality of packets to the outbound queue associated with the outbound port such that a reference to the selected inbound queue storing a plurality of packets is stored in a single one of a plurality of entries in the outbound queue."

Thus, while the Examiner is correct in that the claim does not specifically recite the term "enqueueing," the transferring of an entire queue of packets to a single entry in an outbound queue has the same meaning as "enqueueing an entire queue of packets." Accordingly, Applicant asserts that the Examiner has failed to establish a prima facie case of obviousness.

With respect to the Examiner's assertion that the claims of group (a) were not argued independently and therefore should stand or fall together, Applicant respectfully disagrees. In fact, although a separate paragraph was not dedicated to each, Applicant did specifically address the limitations in claims 2-9 and 21-27. Specifically, claims 1, 2, 22, 3, 4, 5, 23, 6, 24, 7, 25, 8, 26, 9, and 27 were separately addressed, in that order. Accordingly, the patentability of claims 2-

9 and 21-27 should not be solely based upon the patentability of claim 1 or 20, as suggested by the Examiner.

(b) The combination of Erimli, Barucchi and Clark neither discloses nor suggests the invention of claims 45-48, 52

The Examiner again asserts that, with respect to group (b), that the Applicant has failed to argue claims 45-48 and 52 independently and that claims 45-48 and 52 stand or fall together. Applicant respectfully disagrees. In fact, each of the claims 45-48 and 52 are discussed separately, albeit not in separate paragraphs.

Erimli does disclose "one or more entries are passed directly from the output queue write side 76 to the output queue read side." See col. 9, lines 37-47. However, even if more than one entry is "passed," Erimli fails to disclose or suggest transferring an entire inbound queue to a single entry in an outbound queue. In other words, each entry of the outbound queue of Erimli would merely include a single entry from the inbound queue. Thus, Erimli fails to disclose or suggest storing or identifying a single inbound queue in a single entry in an outbound queue. For instance, claim 52 recites "wherein the outbound queue comprises a plurality of entries, each of the plurality of entries storing or identifying one of the multiplicity of inbound queues." The cited art fails to disclose or suggest storing a plurality of inbound queues in an outbound queue in this manner. Accordingly, Applicant respectfully submits that the Examiner has failed to set forth a primary facie case of obviousness.

(c) The combination of Erimli and Barucchi neither discloses nor suggests the invention of claims 10-18, 28-36

On page 15 of the Examiner's answer, the Examiner again asserts that Applicant failed to argue claims 10-18 and 28-36 independently. Applicant again disagrees with this assertion. In fact, claims 10, 28, 11, 29, 12, 30, 13, 31, 14, 32, 15, 33, 16, 34, 17, 35, 18, and 36 were separately addressed in that order, albeit not in separate paragraphs. Applicant did not merely recite what the claims recite, as the Examiner has indicated.

(d) The combination of Erimli, Barucchi and Clark neither discloses nor suggests the invention of claim 51

The Examiner again claims that claim 51 was not argued independently, and is therefore not independently patentable. This is clearly incorrect. Applicant specifically addressed the limitations recited in claim 51, which recites "wherein transferring the inbound queue to a single entry in the outbound queue is performed by the CPU in response to an interrupt" as discussed in Applicant's appeal brief. Accordingly, Applicant respectfully asserts that claim 51 is independently patentable and should not stand or fall with claim 10, from which it is dependent.

It is also important to note that the Examiner cited Clark in this rejection, while Clark was not applied in the rejection of the independent claim 10 upon which claim 51 depends. However, Examiner has failed to point out how Clark is relevant to claim 51. Moreover, the fact that the Examiner cited this third reference implies that claim 51 is not obvious over the combination of Erimli and Barucchi alone. In fact, it appears from the Examiner's rejection of claim 51 that the Examiner cites Clark solely for its disclosure of prior art encryption mechanisms. Since claim 51 does not relate to encryption and Clark does not appear to disclose features that are material to the rejections of claim 51, Applicant respectfully submits that the outstanding rejection of claim 51 under35 USC §103(a) is improper and should be reversed.

(e) The combination of Erimli and Barucchi neither discloses nor suggests the invention of claim 19

The Examiner indicates that Applicant failed to assert that the claims in group (e) stand to fall together. However, as set forth above, this is merely because there is only one claim in group (e) and Applicant believed that this was implied. The Examiner's assertion that claim 19 should stand or fall together with the claims of group (c) seems to be an illogical conclusion. Moreover, Applicant respectfully asserts that the arguments presented were specific to the limitations of claim 19, despite the Examiner's assertion that the remarks are identical to those presented with respect to the claims in group (c). Accordingly, Applicant submits that the patentability of claim

19 should be considered independently from the claims of group (c) as the Examiner has suggested.

(f) The combination of Erimli, Barucchi and Clark neither discloses nor suggests the invention of claims 53 and 54.

Applicant respectfully asserts that since claims 53 and 54 are merely computer-readable medium and apparatus claims, respectively, corresponding to claim 19, discussed above in section (e), and are allowable for the reasons set forth above in section (e). It is unclear why the rejection of claims 53 and 54 differs from that of claim 19. Regardless, Clark fails to cure the deficiencies of Erimli and Barucchi. In fact, it appears that the Examiner cites Clark solely for its disclosure of prior art encryption mechanisms. However, the fact that the Examiner cited this third reference implies that claims 53 and 54 are not obvious over the combination of Erimli and Barucchi alone. Since claims 53 and 54 do not relate to encryption and Clark does not appear to disclose features that are material to the rejections of claims 53 and 54, Applicant respectfully submits that the outstanding rejections of claims 53 and 54 under35 USC §103(a) are improper as set forth above and should be reversed.

(g) The combination of Erimli and Barucchi neither discloses nor suggests the invention of claims 37-42.

The Examiner has again argued that the claims have not been argued individually. However, this is incorrect. Each of claims 37-42 has been addressed separately, albeit not in separate paragraphs. Accordingly, the patentability of each of claims 37-42 should be considered individually, and should not stand or fall together. Moreover, even though the Applicant made reference to arguments previously set forth, these were not the only arguments asserted, as the Examiner has implied. As a result, the claims 37-42 should not stand or fall together with the claims of group (c) or (a) as the Examiner has suggested.

(h) The combination of Erimli, Barucchi and Clark neither discloses nor suggests the invention of claims 43, 49-50

The Examiner asserts that "Clark discusses as prior art encryption mechanisms, which encrypt an entire buffer (queue or set of data packets)" citing column 1, lines 34 through column 2, line 10. While Clark does disclose the encryption of a "buffer," Applicant respectfully submits that the Examiner has mischaracterized the reference by interpreting the term "buffer" to mean a queue or set of data packets. Generally, the term "buffer" is interpreted to mean a data area, while the term "queue" is typically interpreted to include a line or sequence of a plurality of items, which are usually to be handled in a sequential order. In fact, in no manner does Clark disclose or suggest the encryption of an inbound queue **storing a plurality of packets to be separately transmitted**. In fact, claim 43 specifically recites "the outbound controller being adapted for storing a reference to the encrypted inbound queue in a single entry in the one of the plurality of outbound queues, and transmitting data stored in the one of the plurality of outbound queues." Thus, even if Clark were interpreted to teach or suggest the encryption of a queue of packets, Clark fails to cure the deficiencies of the primary references, which fail to teach or suggest transferring a queue of packets to a single entry in an outbound queue. Accordingly, it is respectfully submitted that the outstanding rejection of claim 43 under 35 U.S.C. §103(a) is improper and should be reversed.

(i) Conclusion

In view of the forgoing, it is respectfully submitted that none of the pending claims are anticipated or rendered obvious by any reasonable combination of the art of record. Accordingly, the pending rejection of all of the claims under 35 USC §103 should be reversed.

Respectfully Submitted,
BEYER WEAVER & THOMAS, LLP

Elise R. Heilbrunn
Reg. No. 42,649

Beyer Weaver & Thomas, LLP
P.O. Box 778
Berkeley, California
(510) 843-6200

IX. APPENDIX

CLAIMS ON APPEAL

1. (Previously Amended) A method for providing an inbound controller for a router, the router having an inbound port and an outbound port, a memory, and a CPU, the inbound controller being adapted for receiving an inbound packet at the inbound port, the method comprising:

providing a plurality of inbound queues for the inbound port;

receiving an inbound packet at the inbound port;

classifying the inbound packet in a selected one of the plurality of inbound queues according to packet sorting criteria;

storing the inbound packet in the selected one of the plurality of inbound queues; and

determining when one of the plurality of inbound queues storing a plurality of packets is ready to be moved to an entry in an outbound queue associated with the outbound port, the outbound queue being capable of storing a reference to a multiplicity of inbound queues such that a reference to each of the multiplicity of inbound queues is separately stored in a different one of a plurality of entries in the outbound queue, each of the multiplicity of inbound queues storing a plurality of packets to be separately transmitted.

2. (Previously Amended) The method as recited in claim 1, further including:

asserting an interrupt when it is determined that one of the plurality of inbound queues is ready to be moved to an outbound queue.

3. (Original Claim) The method as recited in claim 1, wherein classifying the inbound packet includes:

selecting inbound packet sorting criteria;

obtaining packet sorting data for the inbound packet, the packet sorting data being associated with the packet sorting criteria; and

sorting the inbound packet into one of the plurality of inbound queues according to the packet sorting data.

4. (Previously Amended) The method as recited in claim 1, the selected one of the plurality of inbound queues corresponding to one of a plurality of outbound queues, the method further comprising:

transferring the selected one of the plurality of inbound queues storing a plurality of packets to the outbound queue associated with the outbound port such that a reference to the selected inbound queue storing a plurality of packets is stored in a single one of a plurality of entries in the outbound queue.

5. (Original Claim) The method as recited in claim 1, wherein storing the inbound packet includes:

obtaining an available packet buffer from a free pool of available packet buffers;

placing the inbound packet in the packet buffer; and

storing the packet buffer in the inbound queue.

6. (Original Claim) The method as recited in claim 1, wherein determining when one of the plurality of inbound queues is ready to be moved to an outbound queue includes:

determining whether a number of packets in one of the plurality of inbound queues exceeds a maximum number of packets.

7. (Original Claim) The method as recited in claim 1, wherein determining when one of the plurality of inbound queues is ready to be moved to an outbound queue includes:

determining whether a number of bytes in one of the plurality of inbound queues exceeds a maximum number of bytes.

8. (Original Claim) The method as recited in claim 1, wherein determining when one of the plurality of inbound queues is ready to be moved to an outbound queue further includes:

determining whether a free pool of available memory has been depleted.

9. (Original Claim) The method as recited in claim 1, wherein determining when one of the plurality of inbound queues is ready to be moved to an outbound queue further includes:

determining whether a maximum time limit has been exceeded.

10. (Previously Amended) A method for providing an outbound controller for a router, the router having an inbound port and an outbound port, a memory, and a CPU, the outbound controller being adapted for forwarding packets at the outbound port, the method comprising:

providing an outbound queue associated with the outbound port and being capable of storing a plurality of inbound queues;

receiving a notification to handle an inbound queue, the inbound queue storing a plurality of packets that are to be separately transmitted;

transferring the inbound queue storing a plurality of packets to a single entry in the outbound queue associated with the outbound port such that a reference to the inbound queue storing a plurality of packets is stored in one of a plurality of entries in the outbound queue; and

repeating the receiving and transferring steps for the plurality of inbound queues such that a reference to each of the plurality of inbound queues is separately stored in a different one of the plurality of entries in the outbound queue.

11. (Original Claim) The method as recited in claim 10, wherein receiving the notification includes:

receiving a notification from the CPU to handle the inbound queue.

12. (Original Claim) The method as recited in claim 10, further including:

transmitting packets stored in the outbound queue.

13. (Original Claim) The method as recited in claim 10, wherein transmitting packets includes:

selectively discarding packets stored in the outbound queue.

14. (Original Claim) The method as recited in claim 10, wherein transmitting packets stored in the outbound queue further includes:

obtaining a next one of the plurality of inbound queues stored in the outbound queue;

transmitting selected packets stored in the next one of the plurality of inbound queues; and

releasing memory associated with the next one of the plurality of inbound queues.

15. (Original Claim) The method as recited in claim 14, wherein releasing the memory includes:

storing the released memory in a free pool of available packet buffers.

16. (Original Claim) The method as recited in claim 14, wherein releasing the memory includes:

forming a new inbound queue to be used by an inbound controller.

17. (Original Claim) The method as recited in claim 14, wherein releasing the memory includes:

forming a queue to be used by the outbound controller during bi-directional operation.

18. (Previously Amended) The method as recited in claim 10, wherein transferring the inbound queue to the outbound queue further includes:

ascertaining a priority of the inbound queue; and

transferring the inbound queue to a single entry in the outbound queue according to the priority of the inbound queue.

19. (Previously Amended) A method for forwarding a packet in a router, the router having a plurality of inbound ports and a plurality of outbound ports, a memory, and a CPU, the method comprising:

providing a plurality of inbound queues for one of the plurality of inbound ports;

providing a plurality of outbound queues, each one of the plurality of outbound queues corresponding to one of the plurality of outbound ports and being capable of storing a plurality of inbound queues;

receiving an inbound packet at the one of the plurality of inbound ports;

classifying the inbound packet in a selected one of the plurality of inbound queues according to packet sorting criteria;

storing the inbound packet in the selected one of the plurality of inbound queues;

repeating the steps of receiving, classifying, and storing until an interrupt is asserted; and

transferring one of the plurality of inbound queues storing a plurality of packets to one of the plurality of outbound queues corresponding to the packet sorting criteria when the interrupt is asserted such that a reference to the one of the plurality of inbound queues storing a plurality of packets is stored in a single one of a plurality of entries in the one of the plurality of outbound queues, wherein each of the plurality of packets in the one of the plurality of inbound queues is to be separately transmitted.

20. (Previously Amended) An inbound controller for a router, the router having an inbound port and an outbound port, a memory, and a CPU, the inbound controller being adapted for receiving an inbound packet at the inbound port, comprising:

a packet receiving module coupled to the inbound port, the packet receiving module being adapted for receiving an inbound packet;

wherein the memory has stored therein:

a plurality of inbound queues for the inbound port;

a classifier adapted for classifying the inbound packet in a selected one of the plurality of inbound queues according to packet sorting criteria;

a packet storing module coupled to the classifier, the packet storing module being adapted for storing the inbound packet in the selected one of the plurality of inbound queues; and

a module adapted for determining when one of the plurality of inbound queues is ready to be moved to an entry in an outbound queue associated with the outbound port, the outbound queue being capable of storing a multiplicity of inbound queues, a reference to each of the multiplicity of inbound queues being stored in a different one of a plurality of entries in the outbound queue, each of the multiplicity of inbound queues storing a plurality of packets that are to be separately transmitted.

21. (Original Claim) The inbound controller as recited in claim 20, further including:

a module adapted for providing the determined one of the plurality of inbound queues.

22. (Original Claim) The inbound controller as recited in claim 20, further including:

a module adapted for asserting an interrupt when it is determined that one of the plurality of inbound queues is ready to be moved by the CPU to the outbound queue.

23. (Original Claim) The inbound controller as recited in claim 20, wherein the packet storing module includes:

a memory obtaining module adapted for obtaining an available packet buffer from a free pool of available packet buffers;

a module adapted for placing the inbound packet in the packet buffer; and

a module adapted for storing the packet buffer in the inbound queue.

24. (Original Claim) The inbound controller as recited in claim 20, wherein the module adapted for determining when one of the plurality of inbound queues is ready to be moved to an outbound queue includes:

a module adapted for determining whether a number of packets in one of the plurality of inbound queues exceeds a maximum number of packets.

25. (Original Claim) The inbound controller as recited in claim 20, wherein the module adapted for determining when one of the plurality of inbound queues is ready to be moved to an outbound queue includes:

a module adapted for determining whether a number of bytes in one of the plurality of inbound queues exceeds a maximum number of bytes.

26. (Original Claim) The inbound controller as recited in claim 20, wherein the module adapted for determining when one of the plurality of inbound queues is ready to be moved to an outbound queue includes:

a module adapted for determining whether a free pool of available memory has been depleted.

27. (Original Claim) The inbound controller as recited in claim 20, wherein the module adapted for determining when one of the plurality of inbound queues is ready to be moved to an outbound queue includes:

a module adapted for determining whether a maximum time limit has been exceeded.

28. (Previously Amended) An outbound controller for a router, the router having an inbound port and an outbound port, a memory, and a CPU, the outbound controller being adapted for forwarding packets at the outbound port, comprising:

a module adapted for receiving a notification to handle an inbound queue associated with the inbound port, the inbound queue storing a plurality of packets;

wherein at least one of the CPU and the memory are adapted for storing an outbound queue associated with the outbound port, the outbound queue being capable of storing a plurality of inbound queues in a plurality of entries in the outbound queue, each of the plurality of inbound queues storing a plurality of packets that are to be separately transmitted; and a queue transferring module adapted for transferring the inbound queue storing a plurality of packets to a single entry in the outbound queue such that a reference to the inbound queue is stored in the entry in the outbound queue.

29. (Original Claim) The outbound controller as recited in claim 28, wherein the module adapted for receiving the notification includes a module adapted for receiving the notification from the CPU.

30. (Original Claim) The outbound controller as recited in claim 28, further including:

a module adapted for transmitting packets stored in the outbound queue.

31. (Original Claim) The outbound controller as recited in claim 30, wherein the module adapted for transmitting packets includes:

a module adapted for selectively discarding packets stored in the outbound queue.

32. (Original Claim) The outbound controller as recited in claim 30, wherein the module adapted for transmitting packets stored in the outbound queue includes:

a module adapted for obtaining a next one of the plurality of inbound queues stored in the outbound queue;

a packet transmission module adapted for transmitting selected packets stored in the next one of the plurality of inbound queues; and

a memory releasing module adapted for releasing memory associated with the next one of the plurality of inbound queues.

33. (Original Claim) The outbound controller as recited in claim 32, wherein the memory releasing module includes:

a module adapted for storing the released memory in a free pool of available packet buffers.

34. (Original Claim) The outbound controller as recited in claim 32, wherein the released memory forms a new inbound queue to be used by an inbound controller.

35. (Original Claim) The outbound controller as recited in claim 32, wherein the released memory forms a queue to be used by the outbound controller during bi-directional operation.

36. (Previously Amended) The outbound controller as recited in claim 28, wherein the queue transferring module is adapted for transferring the inbound queue to a single entry in the outbound queue according to a priority of the inbound queue.

37. (Previously Amended) A router having a plurality of inbound ports and a plurality of outbound ports, a memory, and a CPU, comprising:

an inbound controller coupled to one of the plurality of inbound ports, the inbound controller being adapted for receiving an inbound packet;

wherein the memory has stored therein:

a plurality of inbound queues for the one of the plurality of inbound ports, each one of the plurality of inbound queues being capable of storing a plurality of packets that are to be separately transmitted;

a plurality of outbound queues, each one of the plurality of outbound queues corresponding to one of the plurality of outbound ports and being capable of storing a plurality of inbound queues such that a reference to each of the plurality of inbound queues is stored in a different one of a plurality of entries in the one of the plurality of outbound queues; and

a classifier coupled to the inbound controller, the classifier being adapted for classifying the inbound packet in a selected one of the plurality of inbound queues according to packet sorting criteria, the selected one of the plurality of inbound queues being associated with one of the plurality of outbound queues;

wherein the inbound controller is adapted for storing the inbound packet in the selected one of the plurality of inbound queues.

38. (Previously Amended) The router as recited in claim 37, further including:

an outbound controller coupled to the inbound controller;

wherein the inbound controller selects one of the plurality of inbound queues to be transferred to the outbound controller;

wherein the outbound controller is adapted for storing a reference to the selected one of the plurality of inbound queues in an entry in one of the plurality of outbound queues associated with the packet sorting criteria and transmitting packets stored in the one of the plurality of outbound queues.

39. (Original Claim)The router as recited in claim 37, wherein the inbound controller further includes:

a memory obtaining module coupled to the classifier, the memory obtaining module being adapted for obtaining memory for an inbound packet to permit the inbound packet to be stored in the selected one of the plurality of inbound queues in which the inbound packet is classified.

40. (Original Claim) The router as recited in claim 38, wherein the outbound controller further includes:

a memory releasing module adapted for releasing selected packet buffers associated with packets stored in the one of the plurality of outbound queues.

41. (Original Claim) The router as recited in claim 40, wherein the memory further includes a free pool of available packet buffers and the memory releasing module is adapted for releasing the selected packet buffers into the free pool.

42. (Original Claim) The router as recited in claim 38, wherein the outbound controller further includes:

a memory releasing module adapted for providing a new inbound queue to the inbound controller to replace the selected one of the plurality of inbound queues.

43. (Previously Amended) An encryption system, comprising:

an inbound controller adapted for receiving an inbound packet;

a classifier coupled to the inbound controller and adapted for classifying and storing the inbound packet in one of a plurality of inbound queues;

an outbound controller adapted for receiving the one of the plurality of inbound queues, the one of the plurality of inbound queues storing a plurality of packets to be separately transmitted; and

an encryption box coupled to the outbound controller, the encryption box being adapted for encrypting the one of the plurality of inbound queues to provide an encrypted inbound queue to the outbound controller for transmission, wherein the outbound controller includes an outbound classifier adapted for classifying the encrypted inbound queue in one of a plurality of outbound queues associated with a plurality of outbound ports, the outbound controller being adapted for storing a reference to the encrypted inbound queue in a single entry in the one of the plurality of outbound queues, and transmitting data stored in the one of the plurality of outbound queues.

44. (Cancelled)

45. (Previously Amended) The method as recited in claim 2, further comprising:

when the interrupt is asserted, transferring the one of the plurality of inbound queues to an entry in the outbound queue or an outbound controller associated with the outbound queue.

46. (Previously Amended) The method as recited in claim 45, wherein transferring the one of the plurality of inbound queues to an entry in the outbound queue or an outbound controller associated with the outbound queue is performed by the CPU.

47. (Previously Amended) The method as recited in claim 45, wherein transferring the one of the plurality of inbound queues to an entry in the outbound queue or an outbound controller associated with the outbound queue comprises:

transferring a reference to the one of the plurality of inbound queues to an entry in an outbound queue corresponding to a priority associated with the one of the plurality of inbound queues.

48. (Previously Amended) The method as recited in claim 45, wherein transferring the one of the plurality of inbound queues to the outbound queue or an outbound controller associated with the outbound queue comprises:

transferring a pointer to the one of the plurality of inbound queues to an entry in an outbound queue associated with the one of the plurality of inbound queues.

49. (Original Claim) The method as recited in claim 43, wherein the inbound queue stores therein a plurality of packets, and wherein the encryption box does not encrypt each of the plurality of packets.

50. (Original Claim) The method as recited in claim 43, wherein the encryption box is adapted for encrypting the inbound queue as an entity such that a single encryption step is performed.

51. (Previously Amended) The method as recited in claim 10, wherein transferring the inbound queue to a single entry in the outbound queue is performed by the CPU in response to an interrupt.

52. (Currently Amended) The method as recited in claim 1, wherein the outbound queue comprises a plurality of entries, each of the plurality of entries ~~simultaneously~~ storing or identifying one of the multiplicity of inbound queues.

53. (Previously Amended) A computer-readable medium storing thereon computer-readable instructions for forwarding a packet in a router, the router having a plurality of inbound ports and a plurality of outbound ports, a memory, and a CPU, the method comprising:

instructions for providing a plurality of inbound queues for one of the plurality of inbound ports;

instructions for providing a plurality of outbound queues, each one of the plurality of outbound queues corresponding to one of the plurality of outbound ports and being capable of storing a plurality of inbound queues such that a reference to each of the plurality of inbound queues is simultaneously stored in a different one of a plurality of entries in the one of the plurality of outbound queues;

instructions for receiving an inbound packet at the one of the plurality of inbound ports;

instructions for classifying the inbound packet in a selected one of the plurality of inbound queues according to packet sorting criteria;

instructions for storing the inbound packet in the selected one of the plurality of inbound queues;

instructions for repeating the steps of receiving, classifying, and storing until an interrupt is asserted; and

instructions for transferring one of the plurality of inbound queues to one of the plurality of outbound queues corresponding to the packet sorting criteria when the interrupt is asserted such that a pointer to the transferred one of the plurality of inbound queues is stored in one of a plurality of entries in the one of the plurality of outbound queues, the one of the plurality of inbound queues storing a plurality of packets that are to be separately transmitted.

54. (Previously Amended) An apparatus for forwarding a packet in a router, the router having a plurality of inbound ports and a plurality of outbound ports, a memory, and a CPU, the method comprising:

means for providing a plurality of inbound queues for one of the plurality of inbound ports;

means for providing a plurality of outbound queues, each one of the plurality of outbound queues corresponding to one of the plurality of outbound ports and being capable of storing a plurality of inbound queues such that a reference to each of the plurality of inbound queues is stored in a different one of a plurality of entries in the one of the plurality of outbound queues;

means for receiving an inbound packet at the one of the plurality of inbound ports;

means for classifying the inbound packet in a selected one of the plurality of inbound queues according to packet sorting criteria;

means for storing the inbound packet in the selected one of the plurality of inbound queues;

means for repeating the steps of receiving, classifying, and storing until an interrupt is asserted; and

means for transferring one of the plurality of inbound queues to one of the plurality of outbound queues corresponding to the packet sorting criteria when the interrupt is asserted such that a reference to the transferred one of the plurality of inbound queues is stored in a single one of a plurality of entries in the one of the plurality of outbound queues, the one of the plurality of_ inbound queues storing a plurality of packets that are to be separately transmitted.